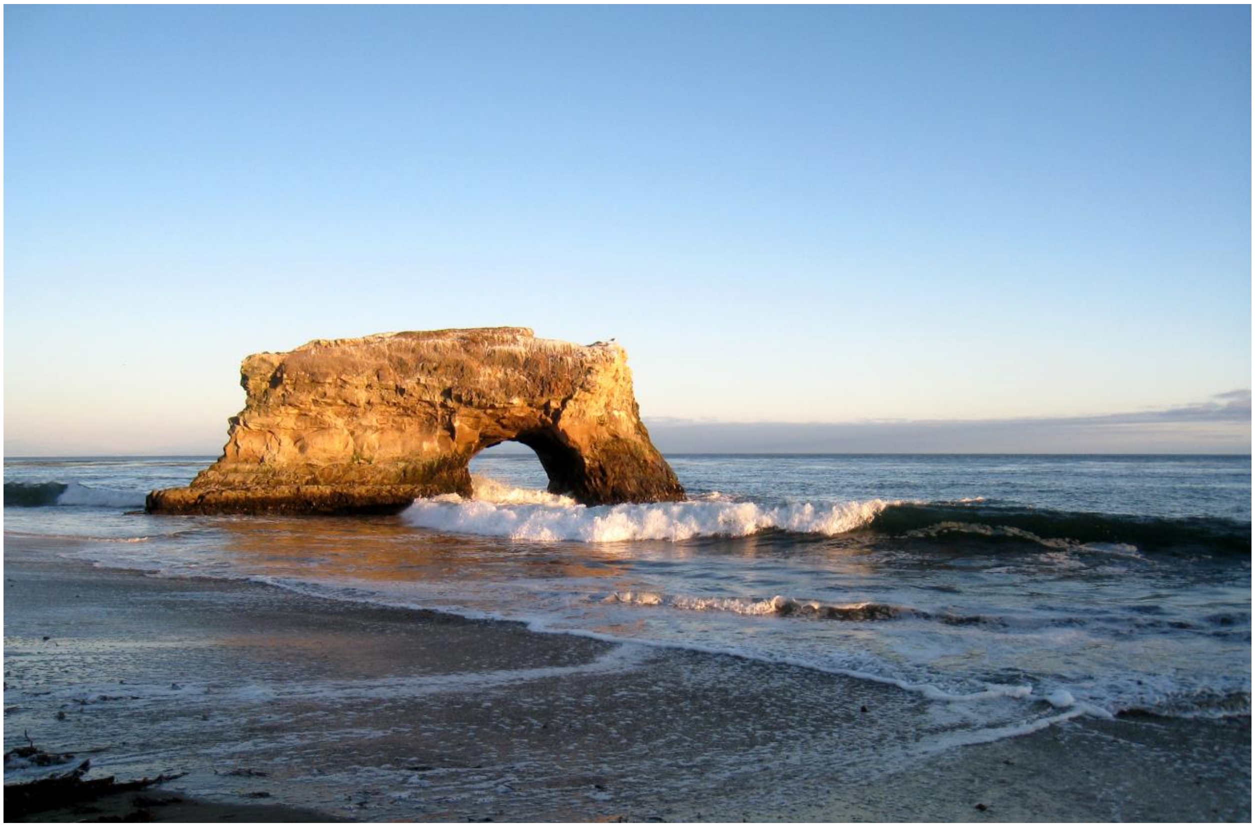
P3: Data Wrangling with MongoDB

By Vaikunth Kannan – Udacity Data Analyst Nanodegree

Area of Study – Santa Cruz, California. This is one of my favourite cities due to the beautiful boardwalk and the kayaking activities there. So, I decided to analyze this place and decided to know more about it.
Open Street Map URL - http://www.openstreetmap.org/relation/111737#map=13/36.9844/-122.0251
Mapzen URl - https://s3.amazonaws.com/metro-extracts.mapzen.com/santa-cruz_california.osm.bz2

References:

1. Udacity Lesson 6 – Data Analysis with MongoDB
2. Zaiste.net
3. http://stackoverflow.com/questions/3095434/inserting-newlines-in-xml-file-generated-via-xml-etree-elementtree-in-python



Problems encountered in the map:

1. Inconsistent tag names
   Mapparser code was run to determine the occurences of each unique element type.
   ```
   {'bounds': 1,
    'member': 4147,
    'nd': 314951,
    'node': 251369,
    'osm': 1,
    'relation': 445,
    'tag': 117035,
    'way': 21138}
   ```

   Further, the keys were investigated to identify if any of the keys are in lower case and those not in lower case would be considered problematic.
   ```
   {'lower': 71765, 'lower_colon': 36473, 'other': 8797, 'problemchars': 0}
   ```

2. To determine the amount of unique users who have contributed to the map of Santa Cruz area, I created a set of users which contained the name of users with unique ids without duplicates. The total number of unique users who contributed to the map dataset was determined to be 436.

3. Street Address Abbreviation:
   One of the problems encountered in this dataset was with street name abbreviation inconsistency. They were corrected using the following map.
   ```
   expected = ["Street", "Avenue", "Boulevard", "Broadway", "Circus", "Close", "Court", "Drive",
   "Court", "Place", "Square", "Lane", "Road",
             "Crescent", "Trail", "Parkway", "Commons", "Garden", "Grove", "Mount", "Park"]

   mapping = {'Ave'  : 'Avenue',
            'Blvd' : 'Boulevard',
            'Dr'   : 'Drive',
            'Ln'   : 'Lane',
            'Pkwy' : 'Parkway',
            'Rd'   : 'Road',
            'Rd.'  : 'Road',
            'St'   : 'Street',
            'street' :"Street",
            'Ct'   : "Court",
            'Cir'  : "Circus",
            'Cr'   : "Court",
            'ave'  : 'Avenue',
            'Sq'   : "Square",
            'Ct'   : "Court",
            'Gdn'  : "Garden",
            'Gr'   : "Grove",
   ```

```
'Pl'   : "Place",
'Cr'   : "Crescent",
'Hwy'  : "Highway",
'Hwy.' : "Highway"}
```

The updated street names were then printed.

```
Chestnut => Chestnut
Mount Hermon Rd => Mount Hermon Road
Rancho Del Mar => Rancho Del Mar
Merrill => Merrill
McAllister WAy => McAllister WAy
Mission Street Extension => Mission Street Extension
Front => Front
front => front
Esplanade => Esplanade
Rodeo Creek Gulch => Rodeo Creek Gulch
245 => 245
Seabright => Seabright
Cedar => Cedar
Pacific => Pacific
Steinhart Way => Steinhart Way
Koshland Way => Koshland Way
Enterprise Way => Enterprise Way
Grace Way => Grace Way
McAllister Way => McAllister Way
Cheryl Way => Cheryl Way
Wolverine Way => Wolverine Way
@ Pasatiempo Sb Ramps => @ Pasatiempo Sb Ramps
Chanticleer Ave => Chanticleer Avenue
41st Ave => 41st Avenue
220 Sylvania Ave => 220 Sylvania Avenue
Fifth Ave => Fifth Avenue
Wilkes Circle => Wilkes Circle
Baskin Circle => Baskin Circle
Engineering Loop => Engineering Loop
```

4. There were a few zipcodes found to be inappropriate and were corrected using a similar code as that used for streets.

```
CA 95062 => 95062
95073 => None
95065-1711 => None
95065 => None
95064 => None
95041 => None
95066 => None
95060 => None
95062 => None
95018 => None
```

```
95066-5121 => None
95062-4205 => None
95010 => None
95003 => None
95002 => None
95066-4024 => None
```

5. In order to transform the XML file to JSON the data.py was used to clean the data and transform it.

Data Overview with MongoDB:

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

1. File size
   - Santa-cruz_california.osm – 52MB
   - Santa-cruz_california.osm.json – 60MB

2. Number of documents

   ```
   > db.santacruz.count()
   272507
   ```

3. Number of ‘nodes’ and ‘ways’

   ```
   > db.santacruz.find({'type':'node'}).count()
   251361
   > db.santacruz.find({'type':'way'}).count()
   21067
   ```

4. Number of unique users

   ```
   db.santacruz.distinct("created.user").length
   430
   ```

5. Top 5 contributing users

   ```
   > db.santacruz.aggregate([{"$group" : {"_id" : "$created.user", "count" : {"$sum" : 1}}}, {"$sort" : {"count" : -1} }, {"$limit" : 5}])
   { "_id" : "stevea", "count" : 154616 }
   { "_id" : "nmixter", "count" : 41068 }
   { "_id" : "DanHomerick", "count" : 25577 }
   { "_id" : "adelman", "count" : 5257 }
   { "_id" : "woodpeck_fixbot", "count" : 4415 }
   ```

6. Number of users with only one post

   ```
   db.santacruz.aggregate([{"$group" : {"_id" : "$created.user", "count" : {"$sum" : 1}}}, {"$group" : {"_id" : "$count", "num_users" : {"$sum" : 1}}}, {"$sort" : {"_id" : 1}}, {"$limit" : 1}]);

   { "_id" : 1, "num_users" : 91 }
   ```

7. Most common building types

   ```
   > db.santacruz.aggregate([{'$match' : {'building' : {'$exists' : 1}}}, {'$group' : {'_id' : '$building',
   'count' : {'$sum' : 1}}}, {'$sort' : {'count' : -1}}, {'$limit' : 5}]);
   { "_id" : "yes", "count" : 3850 }
   { "_id" : "commercial", "count" : 142 }
   { "_id" : "house", "count" : 131 }
   { "_id" : "residential", "count" : 123 }
   { "_id" : "apartments", "count" : 114 }
   ```

8. List of top amenities in Santa Cruz

   ```
   > db.santacruz.aggregate([{"$match": {"amenity" : {"$exists" : 1}}}, {"$group" : {"_id" :
   "$amenity", "count" : {"$sum" : 1}}},
   ... {"$sort" : {"count" : -1}}, {"$limit" : 10}]);
   { "_id" : "parking", "count" : 951 }
   { "_id" : "bicycle_parking", "count" : 263 }
   { "_id" : "restaurant", "count" : 249 }
   { "_id" : "toilets", "count" : 232 }
   { "_id" : "bench", "count" : 201 }
   { "_id" : "place_of_worship", "count" : 141 }
   { "_id" : "school", "count" : 105 }
   { "_id" : "recycling", "count" : 96 }
   { "_id" : "cafe", "count" : 91 }
   { "_id" : "fast_food", "count" : 68 }
   ```

9. List of top types of cuisines in restaurants.

   ```
   > db.santacruz.aggregate([{"$match": {"amenity" : {"$exists" : 1},"amenity" : "restaurant",}},
   {"$group" : {"_id" : {"Food" : "$cuisine"}, "count" : {"$sum" : 1}}},{"$project" : {"_id" : 0, "Food" :
   "$_id.Food", "Count" : "$count"}}, {"$sort" : {"count" : -1}}, {"$limit" : 6}]);
   { "Food" : "kebab", "Count" : 1 }
   { "Food" : "american", "Count" : 8 }
   { "Food" : "continental", "Count" : 1 }
   { "Food" : "indian", "Count" : 2 }
   { "Food" : null, "Count" : 80 }
   { "Food" : "regional", "Count" : 7 }
   ```

10. Most common street address in the dataset

    ```
    >  db.santacruz.aggregate([{'$match' : {'address.street' : {'$exists' : 1}}}, {'$group' : {'_id' :
    '$address.street', 'count' : {'$sum' : 1}}}, {'$sort' : {'count' : -1}}, {'$limit' : 1}]);

    { "_id" : "Porter-Kresge Road", "count" : 35 }
    ```

11. Nodes without addresses

    ```
    > db.santacruz.aggregate([{'$match' : {'type' : 'node', 'address' : {'$exists' : 0}}}, {'$group' : {'_id' :
    'Nodes without addresses', 'count' : {'$sum' : 1}}}]);
    { "_id" : "Nodes without addresses", "count" : 251022 }
    ```

Conclusion:

With the data analysis using MongoDB, we can see that there was more emphasis on 'nodes' rather than 'ways'. There is a large number of nodes approximately 25k which are without addresses around 90% of the whole dataset. An additional idea to this dataset would be compress this data by removing way tags and use only node tags. This would reduce the size of the database. Further, if ways are needed, then we could remove the improper nodes that were referenced in 'relation' and 'member' tags.

There are still more opportunities to clean the data and validate it. The most recent data for Open Street Map was around 2005, so more data would be needed to keep it up to date or we could use google maps api to analyze the data and find inappropriate values.

Also, point of interests could be added in the dataset by using data from websites like FourSquare which is a local search and discovery service.